**Getting started with R Studio**

This is a basic guide on how to connect and get started using the IRIS Warehouse through R Studio.

**Helpful resources:**

<https://db.rstudio.com/getting-started/>

Packages to download:

dplyr + dbplyr

DBI

Odbc

The main way to connect to the database is using the code snippet below. Using Trusted\_Connection makes Windows Authentication possible.

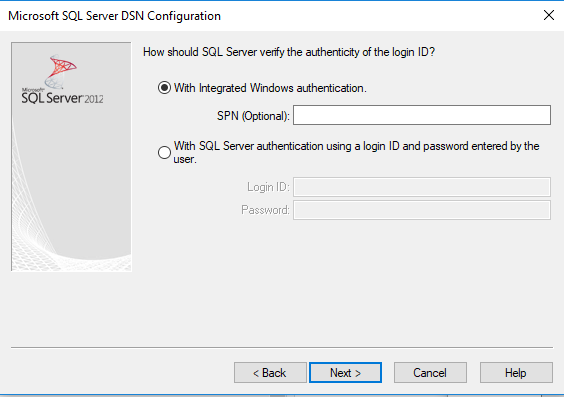
1. library(DBI)
2. library(odbc)
3. con <- dbConnect(odbc(),
4. Driver = "ODBC Driver 17 for SQL Server",
5. Server = "ifw9ecos-bvdb11.fws.doi.net\\ecos\_beta",
6. Database = "IRIS\_DataWarehouse",
7. Trusted\_Connection = "yes",
8. Port = 1433)

Another way to set up a connection to the SQL Server with ODBC is to set it through **ODBC Data Sources.** Go to Windows search and type “ODBC Data Sources”. If this does not appear, the necessary packages were not downloaded.

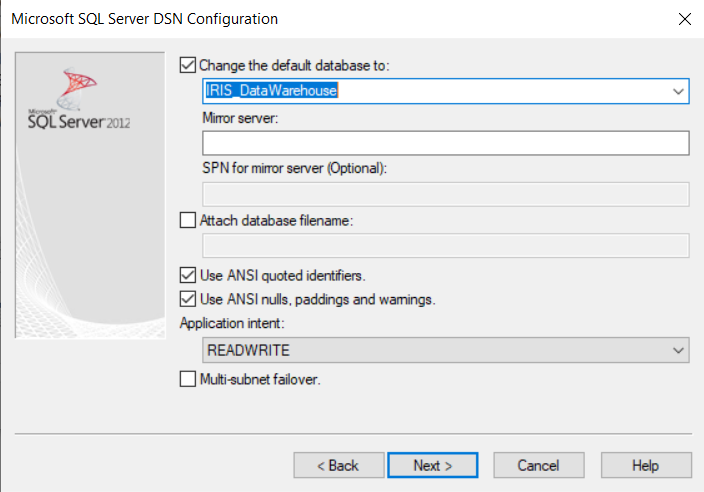
Click “Add” 🡪 Select “ODBC Driver 17 for SQL Server” 🡪 Finish

Name it IRIS\_DataWarehouse and use the server: ifw9ecos-bvdb11.fws.doi.net\ecos\_beta

Hit **Next** and **make sure the connection is set to “With Integrated Windows authentication”**



Hit Next and set the database to “IRIS\_DataWarehouse”.



Keep hitting Next and Finish. The test Data Source should come in as “Successful”. The data source is now available to use.

The data source can now be called as follows:

library(DBI)

con <- dbConnect(odbc::odbc(), "IRIS\_DataWarehouse")

There are 3 main methods used for SQL queries in R Studio:

* 1. Using R Notebook, you can paste direct SQL queries as chunks
  2. With the DBI library, you can paste SQL queries using the dbGetQuery() function
  3. Use the dbplyr and dplyr libraries with their own formatting

The warehouse is organized using dimension and fact tables. The dimension table provides commonly used descriptive/qualitative information recorded by fact tables. It contains the attributes of the measuerments found in fact tables. Fact tables can include foreign keys that can be joined back to dimensions to answer the set of needed questions. For more information, refer back to the main IRIS User Guide.

Also, check to see the warehouse status to see when the last sync was made with sources.

Status <- dBGetQuery(con, ‘SELECT \* FROM dbo.\_WarehouseStatus’)

or for a specific table

status <- dBGetQuery(con, ‘SELECT \* FROM dbo.\_WarehouseStatus WHERE TableName = \’FactSurveyInventory\’ ‘)

**1. R Notebook Example with SQL Chunk**

Using R notebook, it’s possible to save a SQL chunk as an object and use that to make calls.

Make two connection chunks. One chunk for the database connection and the other to paste SQL code into.

First chunk:

1. ```{r setup, include=FALSE}
3. library(DBI)
4. library(odbc)
5. con <- dbConnect(odbc(),
6. Driver = "ODBC Driver 17 for SQL Server",
7. Server = "ifw9ecos-bvdb11.fws.doi.net\\ecos\_beta",
8. Database = "IRIS\_DataWarehouse",
9. Trusted\_Connection = "yes",
10. Port = 1433)
11. knitr::opts\_chunks$set(connection = “con”)
12. ```

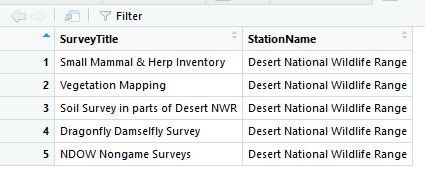
SQL Chunk:

1. ```{sql, **output**.var="markup\_test"}

4. **SELECT**
5. **TOP** 10
6. s.SurveyName **AS** SurveyTitle
7. ,o.FullName **AS** StationName
8. **FROM**
9. IRIS\_DataWarehouse.Refuges.FactSurveyInventory F
10. **INNER** JOIN IRIS\_DataWarehouse.Refuges.DimSurvey s **ON** F.SurveyId=s.ID
11. **INNER** JOIN IRIS\_DataWarehouse.dbo.DimOrganization o **ON** F.Station\_OrganizationID = o.ID
12. **WHERE**
13. s.SurveyType = 'Inventory' AND o.FullName = 'Desert National Wildlife Range'

16. ```

Running the query above gives the example result below (markup\_test) :



**2. QUERY WITH DBI**

It’s also possible to paste SQL code directly into a DBI function called dbGetQuery()as a quoted string and save it as variable.

1. ```{r}
2. library(odbc)
3. library(dplyr)
4. #Database connection
5. con <- dbConnect(odbc::odbc(), "IRIS\_DataWarehouse")

8. query1 <- dbGetQuery(con, '**SELECT**
9. **TOP** 10
10. s.SurveyName **AS** SurveyTitle
11. ,o.FullName **AS** StationName
12. **FROM**
13. IRIS\_DataWarehouse.Refuges.FactSurveyInventory F
14. **INNER** JOIN IRIS\_DataWarehouse.Refuges.DimSurvey s **ON** F.SurveyId=s.ID
15. **INNER** JOIN IRIS\_DataWarehouse.dbo.DimOrganization o **ON** F.Station\_OrganizationID = o.ID
16. **WHERE**
17. s.SurveyType = \'Inventory\' AND o.FullName = \'Desert National Wildlife Range\'')

Above is a long example, but queries can be broken up.

1. #query fact table
2. fct\_survey <- dbGetQuery(con, 'SELECT \* FROM Refuges.FactSurveyInventory')
3. #dimension table
4. dim\_survey <- dbGetQuery(con, 'SELECT ID, SurveyIdNumber, SurveyName, SurveyType FROM Refuges.DimSurvey')

**3. Query using dplyr + dblyr**

1. library(odbc)
2. library(dplyr)
3. library(dbplyr)
4. #Database connection
5. con <- dbConnect(odbc::odbc(), "IRIS\_DataWarehouse”)

8. fct\_survey <- tbl(con, in\_schema("Refuges", "FactSurveyInventory"))
10. dim\_survey <- tbl(con, in\_schema("Refuges", "DimSurvey"))
12. q\_fct <- fct\_survey %>%
14. **select**(ID, SurveyId)
16. q\_dim <- dim\_survey %>%
17. **select**(ID, SurveyIdNumber, SurveyName, SurveyPriority, SurveyType)
19. #the head() **function** can be used **to** **check** the **first** 10 **rows** **of** the **table**
20. #Check top 10 facts
21. top\_10\_fct <- head(q\_fct, n = 10)
23. #top\_10\_dim <- head(q\_dim, n = 10)
25. #filter **by** surveytype = inventory
26. inventory\_list <- filter(q\_dim, SurveyType == "Inventory")
28. #joins
30. join\_query <- q\_fct %>%
31. inner\_join(q\_dim, **by** = c("SurveyId"="ID" ))

More on join formatting

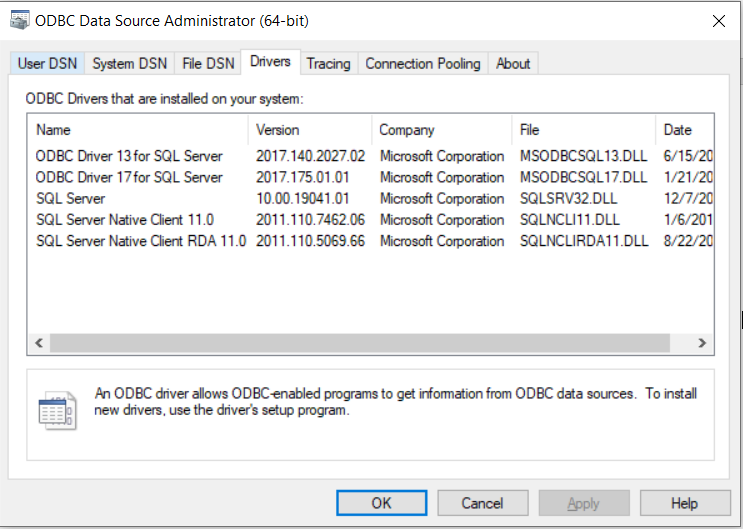
1. fct\_survey2 <- tbl(con, in\_schema("Refuges", "FactSurveyInventory"))
3. dim\_survey2 <- tbl(con, in\_schema("Refuges", "DimSurvey"))

6. dim\_survey2 %>%
7. **select**(ID, SurveyIdNumber, SurveyName, SurveyPriority, SurveyType)
9. join\_query <- fct\_survey2 %>%
11. **select**(ID, SurveyId) %>%
13. inner\_join(dim\_survey2, **by** = c("SurveyId"="ID" ))

**4.Troubleshooting**

Make sure all SQL drivers exist on your computer. Double check in R or by going to Search 🡪 odbc data sources (64-bit). Some computers will not have the ODBC Driver 17 downloaded. Download it here:

<https://docs.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server?view=sql-server-ver15>



You can also switch over to the “SQL Server Native Client 11.0” if the driver is up to date to connect or try another ODBC driver.

There is also a bug in ODBC when joining fact and dimension tables. If a table has a mix of long data columns, the longest data column needs to be at the end of the select list. The dimension columns cannot be retrieved in arbitrary order.